import random

import string

ALPHABET = string.ascii\_letters

NUMERIC = string.digits

SYMBOLIC = string.punctuation

def encrypt\_password(password):

password\_characters = [char for char in password]

key = str(random.randint(111, 999))

alphabet\_key, numeric\_key, symbolic\_key = int(key[0]), int(key[1]), int(key[2])

encrypted\_password = []

for char in password\_characters:

if char in ALPHABET:

alphabet\_index = ALPHABET.index(char)

new\_char = (alphabet\_index + alphabet\_key) % len(ALPHABET)

encrypted\_password.append(ALPHABET[new\_char])

elif char in NUMERIC:

numeric\_index = NUMERIC.index(char)

new\_char = (numeric\_index + numeric\_key) % len(NUMERIC)

encrypted\_password.append(NUMERIC[new\_char])

elif char in SYMBOLIC:

symbolic\_index = SYMBOLIC.index(char)

new\_char = (symbolic\_index + symbolic\_key) % len(SYMBOLIC)

encrypted\_password.append(SYMBOLIC[new\_char])

return ''.join(encrypted\_password), key

def decrypt\_password(encrypted\_password):

decrypted\_password = []

alphabet\_key, numeric\_key, symbolic\_key = int(encrypted\_password[1][0]), int(encrypted\_password[1][1]), int(

encrypted\_password[1][2])

for char in encrypted\_password[0]:

if char in ALPHABET:

alphabet\_index = ALPHABET.index(char)

new\_char = (alphabet\_index - alphabet\_key) % len(ALPHABET)

decrypted\_password.append(ALPHABET[new\_char])

elif char in NUMERIC:

numeric\_index = NUMERIC.index(char)

new\_char = (numeric\_index - numeric\_key) % len(NUMERIC)

decrypted\_password.append(NUMERIC[new\_char])

elif char in SYMBOLIC:

symbolic\_index = SYMBOLIC.index(char)

new\_char = (symbolic\_index - symbolic\_key) % len(SYMBOLIC)

decrypted\_password.append(SYMBOLIC[new\_char])

return ''.join(decrypted\_password)

def generate\_password():

num\_of\_alphabets = random.randint(5, 7)

num\_of\_numbers = random.randint(3, 5)

num\_of\_symbols = random.randint(1, 3)

password = [random.choice(ALPHABET) for \_ in range(num\_of\_alphabets)]

password.extend([random.choice(NUMERIC) for \_ in range(num\_of\_numbers)])

password.extend([random.choice(SYMBOLIC) for \_ in range(num\_of\_symbols)])

random.shuffle(password)

return ''.join(password)

def create\_file(name, password\_amount=1, password=''):

with open(f'{name}.txt', 'w+') as f:

if password == '':

for i in range(password\_amount):

encrypted\_password = encrypt\_password(generate\_password())

password = encrypted\_password[0]

key = encrypted\_password[1]

f.write(f'\nPassword {i + 1}: {password} || Key {i + 1}: {key}')

else:

for i in range(password\_amount):

encrypted\_password = encrypt\_password(password)

f.write(f'\nPassword {i + 1}: {encrypted\_password[0]} || Key {i + 1}: {encrypted\_password[1]}')

def decrypt\_file(file):

with open(f'{file}.txt', 'r') as f:

lines = f.readlines()

lines.pop(0)

for i, line in enumerate(lines):

password\_info = line.split('||')

password = password\_info[0].split(':')[1].strip()

key = password\_info[1].split(':')[1].strip()

decrypted = decrypt\_password([password, key])

print(f'{i + 1}. The password ({password}) was decrypted into: ({decrypted})')

# Example usage:

create\_file('passwords', 5) # Creates a file 'passwords.txt' with 5 encrypted passwords

decrypt\_file('passwords') # Decrypts the passwords from the file